**Final Project**

**Bahamas Sports Physio Center**

By

Jonathan Del Corpo - 40031689

Ahnaf Shahriar - 40025207

Yeashin Rabbi - 26789382

Ali At-Tayeb - 27732929

COMP 353

Dr. Khaled Jababo

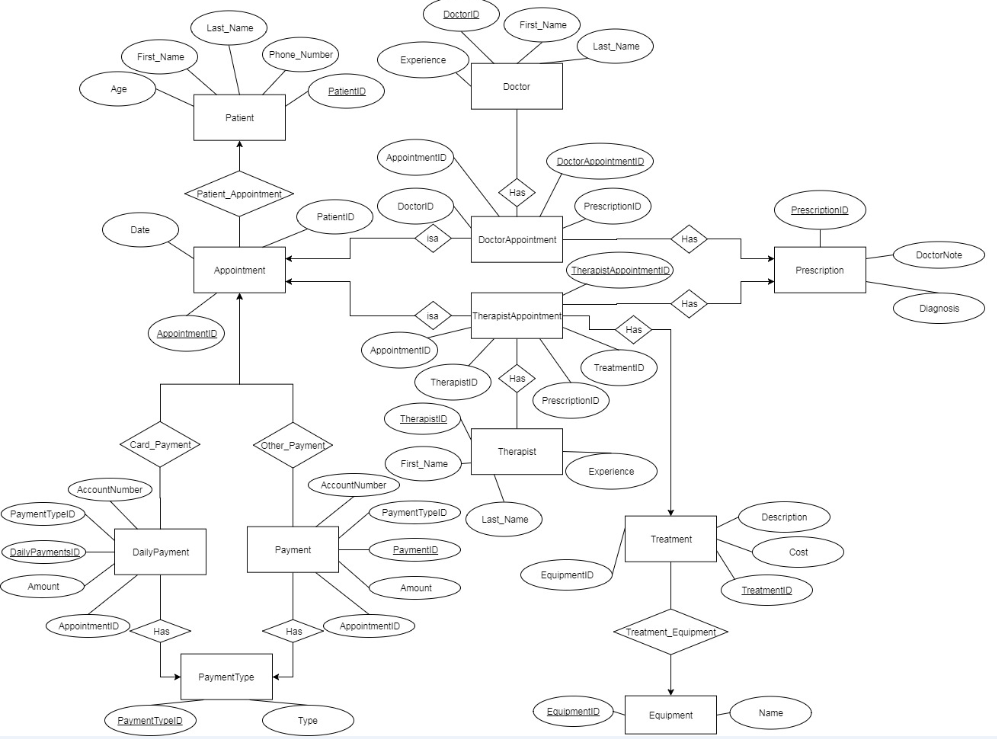
Concordia University

August 8th 2017

**Table of Contents**

1. ER Diagram……………………………………………………………………………… 3
2. Assumptions……………………………………………………………….…………… 4
3. Relational Schema ………………………………………………………………….. 5
4. 3NF Conversion ………………………………………………………………………..5
5. SQL Codes……………………………………………………………………………….. 6
6. Triggers……………………………………………………………………………………11
7. User-Interface………………………………………………………………………… 12
8. Contributions…………………………………………………………………………. 22
9. Sources…………………………………………………………………………………...23

**ER DIAGRAM**

****

**ASSUMPTIONS**

* Patients, doctors, therapist, receptionists and nurses all have a UserID attribute. The user id holds their username and password to log in the system.
* All appointments have an appointment id which is a foreign key to both DoctorAppointment and Therapist Appointment.
* A doctor can only assign prescriptions.
* A therapist can assign prescriptions and provide treatment.
* PaymentType holds 4 values; Cash, Cheque, Debit, Credit
* A treatment can have one equipment linked to it. The same equipment can be used for many treatments. A treatment also does not necessarily require an equipment.

-If the patient decides to pay using a debit or credit card, the payment goes into DailyPayment.

-If the patient decides to pay using cash or cheque, the payment goes into Payment.

-Debit, Credit and cheques require an account number.

-Every day at 5pm, the DailyPayment table is sent to the bank institution and the values and copied to the Payment Table.

-Every day at 5:05pm, the DailyPayment table is truncated to be empty on the following day when the center opens.

**RELATIONAL SCHEMA**

Users (UserID, AccessRightsID, UserName, Password)

AccessRights (AccessRightsID, Name, AccessLevel)

UserInformation (UserInformationID, UserID, First\_Name, Last\_Name, Phone\_Number, Age)

Doctor (DoctorID, UserID, Experience)

Receptionist (ReceptionistID, UserID)

Patient (PatientID, UserID)

Therapist (TherapistID, UserID, Experience)

Nurse (NurseID, UserID)

Treatment (TreatmentID, EquipmentID, Description, Cost)

Equipment (EquipmentID, Name)

Prescription (Prescription ID, DoctorsNote, Diagnosis)

Appointment (AppointmentID, PatientID, Appointment\_Date)

TherapistAppointment (TherapistAppointmentID, AppointmentID, TherapistID, PrescriptionID,

TreatmentID)

Doctor Appointment (DoctorAppointmentID, AppointmentID, PrescriptionID, DoctorID)

DailyPayment (DailyPaymentID, PaymentTypeID, AppointmentID, Amount, AccountNumber)

Payment (PaymentID, PaymentTypeID, AppointmentID, Amount, AccountNumber)

PaymentType (PaymentTypeID, Type)

**3NF CONVERSION**

Our database tables are in 3NF because all our FDs are trivial. The LHS of all FDs are super keys. The RHS is part of a super key. If we were to check our tables we wouldn’t find any inconsistent or duplicated data. This is achieved by implementing 3NF

**SQL CODES**

CREATE TABLE User (

UserID INT (8) UNSIGNED AUTO\_INCREMENT,

AccessRightsID INT (8) UNSIGNED,

Username VARCHAR (30) UNIQUE,

Password VARCHAR (95),

PRIMARY KEY (UserID));

ALTER TABLE User

ADD CONSTRAINT FK\_User\_AccessRights

FOREIGN KEY (AccessRightsID)

REFERENCES AccessRights (AccessRightsID);

CREATE TABLE AccessRights (

AccessRightsID INT (8) UNSIGNED AUTO\_INCREMENT,

Name VARCHAR (30),

AccessLevel INT (2),

PRIMARY KEY (AccessRightsID));

INSERT INTO AccessRights VALUES (0, "Patient", 1);

INSERT INTO AccessRights VALUES (0, "Nurse", 2);

INSERT INTO AccessRights VALUES (0, "Therapist", 2);

INSERT INTO AccessRights VALUES (0, "Doctor", 2);

INSERT INTO AccessRights VALUES (0, "Receptionist", 3);

CREATE TABLE UserInformation (

UserInformationID INT (8) UNSIGNED AUTO\_INCREMENT,

UserID INT (8) UNSIGNED,

First\_Name VARCHAR (30),

Last\_Name VARCHAR (30),

Phone\_Number VARCHAR (10),

Age INT (3) UNSIGNED,

PRIMARY KEY (UserInformationID),

CHECK (Age>=18));

ALTER TABLE UserInformation

ADD CONSTRAINT FK\_UserInformation\_User

FOREIGN KEY (UserID)

REFERENCES User (UserID)

ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE Doctor (

DoctorID INT (8) UNSIGNED AUTO\_INCREMENT,

UserID INT (8) UNSIGNED,

Experience INT (2),

PRIMARY KEY (DoctorID),

CHECK (Experience>=6));

ALTER TABLE Doctor

ADD CONSTRAINT FK\_Doctor\_User

FOREIGN KEY (UserID)

REFERENCES User (UserID)

ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE Therapist (

TherapistID INT (8) UNSIGNED AUTO\_INCREMENT,

UserID INT (8) UNSIGNED,

Experience INT (2),

PRIMARY KEY (TherapistID),

CHECK (Experience>=2));

ALTER TABLE Therapist

ADD CONSTRAINT FK\_Therapist\_User

FOREIGN KEY (UserID)

REFERENCES User (UserID)

ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE Patient (

PatientID INT (8) UNSIGNED AUTO\_INCREMENT,

UserID INT (8) UNSIGNED,

PRIMARY KEY (PatientID));

ALTER TABLE Patient

ADD CONSTRAINT FK\_Patient\_User

FOREIGN KEY (UserID)

REFERENCES User (UserID)

ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE Receptionist (

ReceptionistID INT (8) UNSIGNED AUTO\_INCREMENT,

UserID INT (8) UNSIGNED,

PRIMARY KEY (ReceptionistID));

ALTER TABLE Receptionist

ADD CONSTRAINT FK\_Receptionist\_User

FOREIGN KEY (UserID)

REFERENCES User (UserID)

ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE Nurse (

NurseID INT (8) UNSIGNED AUTO\_INCREMENT,

UserID INT (8) UNSIGNED,

PRIMARY KEY (NurseID));

ALTER TABLE Nurse

ADD CONSTRAINT FK\_Nurse\_User

FOREIGN KEY (UserID)

REFERENCES User (UserID)

ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE Treatment (

TreatmentID INT (8) UNSIGNED AUTO\_INCREMENT,

EquipmentID INT (8) UNSIGNED,

Description VARCHAR (50),

Cost DOUBLE (10, 2),

PRIMARY KEY (TreatmentID));

ALTER TABLE Treatment

ADD CONSTRAINT FK\_Treatment\_Equipment

FOREIGN KEY (EquipmentID)

REFERENCES Equipment (EquipmentID);

CREATE TABLE Equipment (

EquipmentID INT (8) UNSIGNED AUTO\_INCREMENT,

Name VARCHAR (30),

PRIMARY KEY (EquipmentID));

CREATE TABLE Prescription (

PrescriptionID INT (8) UNSIGNED AUTO\_INCREMENT,

DoctorsNote VARCHAR (1000),

Diagnosis VARCHAR (50),

PRIMARY KEY (PrescriptionID));

CREATE TABLE Appointment (

AppointmentID INT (8) UNSIGNED AUTO\_INCREMENT,

PatientID INT (8) UNSIGNED,

Appointment\_Date Date,

PRIMARY KEY (AppointmentID));

ALTER TABLE Appointment

ADD CONSTRAINT FK\_Appoitment\_Patient

FOREIGN KEY (PatientID)

REFERENCES Patient (PatientID)

ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE TherapistAppointment (

TherapistAppointmentID INT (8) UNSIGNED AUTO\_INCREMENT,

AppointmentID INT (8) UNSIGNED,

TherapistID INT (8) UNSIGNED,

PrescriptionID INT (8) UNSIGNED,

TreatmentID INT (8) UNSIGNED,

PRIMARY KEY (TherapistAppointmentID));

ALTER TABLE TherapistAppointment

ADD CONSTRAINT FK\_Therapist\_Appointment

FOREIGN KEY (AppointmentID)

REFERENCES Appointment (AppointmentID)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE TherapistAppointment

ADD CONSTRAINT FK\_Appointment\_Therapist

FOREIGN KEY (TherapistID)

REFERENCES Therapist (TherapistID)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE TherapistAppointment

ADD CONSTRAINT FK\_Therapist\_Appointment\_Prescription

FOREIGN KEY (PrescriptionID)

REFERENCES Prescription (PrescriptionID);

ALTER TABLE TherapistAppointment

ADD CONSTRAINT FK\_Therapist\_Appointment\_Treatment

FOREIGN KEY (TreatmentID)

REFERENCES Treatment (TreatmentID);

CREATE TABLE DoctorAppointment (

DoctorAppointmentID INT (8) UNSIGNED AUTO\_INCREMENT,

AppointmentID INT (8) UNSIGNED,

DoctorID INT (8) UNSIGNED,

PrescriptionID INT (8) UNSIGNED,

PRIMARY KEY (DoctorAppointmentID));

ALTER TABLE Doctor Appointment

ADD CONSTRAINT FK\_Doctor\_Appointment

FOREIGN KEY (AppointmentID)

REFERENCES Appointment (AppointmentID)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE DoctorAppointment

ADD CONSTRAINT FK\_Appointment\_Doctor

FOREIGN KEY (DoctorID)

REFERENCES Doctor (DoctorID)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE DoctorAppointment

ADD CONSTRAINT FK\_Doctor\_Appointment\_Prescription

FOREIGN KEY (PrescriptionID)

REFERENCES Prescription (PrescriptionID);

CREATE TABLE DailyPayment (

DailyPaymentID INT (8) UNSIGNED AUTO\_INCREMENT,

PaymentTypeID INT (8) UNSIGNED,

AppointmentID INT (8) UNSIGNED,

Amount DOUBLE (10, 2),

AccountNumber VARCHAR (16),

PRIMARY KEY (DailyPaymentID));

ALTER TABLE DailyPayment

ADD CONSTRAINT FK\_PaymentType\_DailyPayment

FOREIGN KEY (PaymentTypeID)

REFERENCES PaymentType (PaymentTypeID);

ALTER TABLE DailyPayment

ADD CONSTRAINT FK\_Appointment\_DailyPayment

FOREIGN KEY (AppointmentID)

REFERENCES Appointment (AppointmentID)

ON UPDATE CASCADE;

CREATE TABLE Payment (

PaymentID INT (8) UNSIGNED AUTO\_INCREMENT,

PaymentTypeID INT (8) UNSIGNED,

AppointmentID INT (8) UNSIGNED,

Amount DOUBLE (10, 2),

AccountNumber VARCHAR (16),

PRIMARY KEY (PaymentID));

CREATE TABLE PaymentType (

PaymentTypeID INT (8) UNSIGNED AUTO\_INCREMENT,

Type ENUM ('Cash','Cheque','Debit','Credit'),

PRIMARY KEY (PaymentTypeID));

INSERT INTO PaymentType values (0,"Cash");

INSERT INTO PaymentType values (0,"Cheque");

INSERT INTO PaymentType values (0,"Debit");

INSERT INTO PaymentType values (0,"Credit");

ALTER TABLE Payment

ADD CONSTRAINT FK\_PaymentType\_Payment

FOREIGN KEY (PaymentTypeID)

REFERENCES PaymentType (PaymentTypeID);

ALTER TABLE Payment

ADD CONSTRAINT FK\_Appointment\_Payment

FOREIGN KEY (AppointmentID)

REFERENCES Appointment (AppointmentID)

ON UPDATE CASCADE;

**TRIGGERS**

CREATE

EVENT `Daily\_Payment\_Verification`

ON SCHEDULE EVERY 1 DAY STARTS '2017-08-08 17:00:00' ON COMPLETION NOT PRESERVE ENABLE

DO

INSERT INTO Payment (PaymentTypeID, AppointmentID, Amount, AccountNumber)

SELECT PaymentTypeID, AppointmentID, Amount, AccountNumber

FROM DailyPayment;

CREATE

EVENT `Delete\_Daily\_Payment`

ON SCHEDULE EVERY 1 DAY STARTS '2017-08-08 17:05:00' ON COMPLETION NOT PRESERVE ENABLE

DO

TRUNCATE DailyPayment;

**USER-INTERFACE**

To test out the system, a couple of users have been created. Users can have the password of their choice. For simplicity sake, the following users have “12345” as their password. They can be used to test the functionalities of the system.

|  |  |
| --- | --- |
| **User type** | **Username** |
| Receptionists | Jonguy |
| Ahnaf |
| Patients | JuliaM |
| MarkA |
| TristanB |
| ManonG |
| Doctors | MustaphaH |
| RoxanneL |
| Therapists | SandraH |
| SaadM |
| Nurses | LauraL |
| LamiceL |

**Login Page:** This is the page that is presented to the user when they try to access our system.

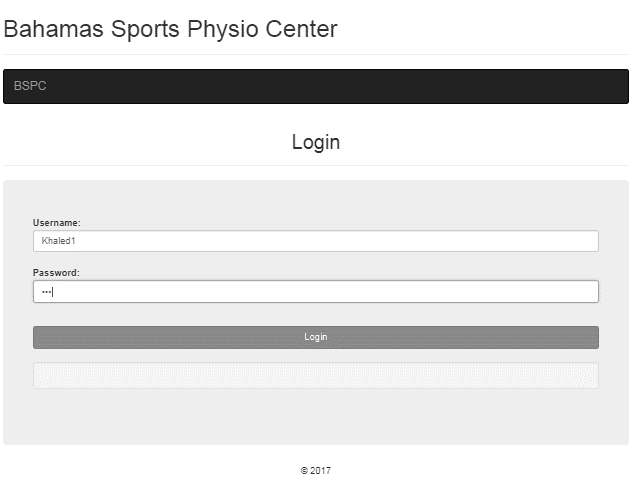


Figure 1: Login Page

**Home Page:** When a user logs in his/her is brought to this page

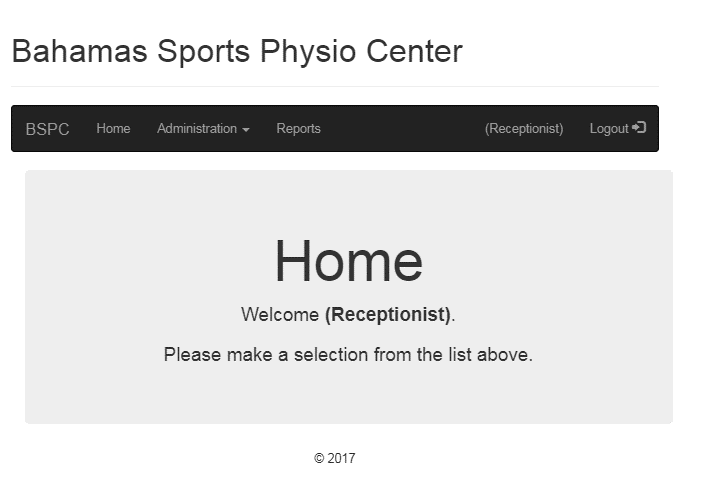


Figure 2: Home Page

**Registration:** This is the page where the receptionist can register a user(patient/employee) to the system. We can also register an employee into the system.

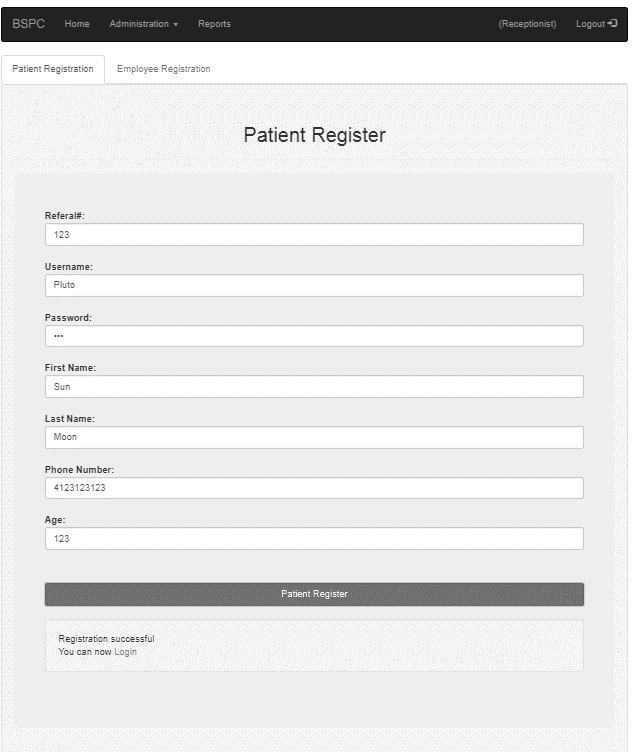


Figure 3: Registration Page

**Appointment Page**: This is the page where a receptionist books an appointment for a user. There is a tab for booking an appointment with the doctor and another one for booking an appointment with a therapist. They both look the same. An appointment cannot be booked from a past date and it to be booked 8 weeks prior to the appointment date. We consider 8 weeks to be 56 days

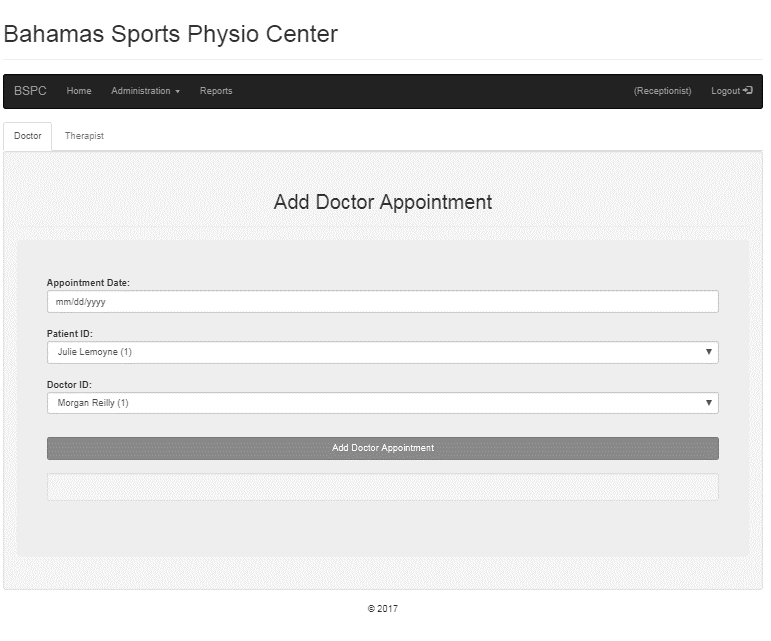


Figure 4: Appointment Page

**Patients List page**: This is the page that shows all the relevant information about patients.

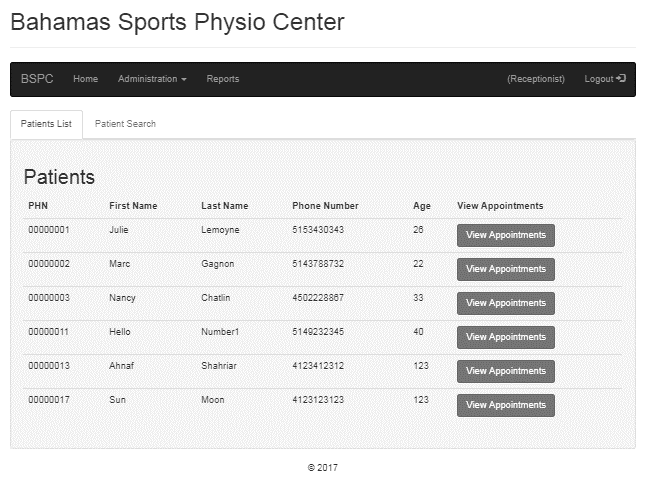


Figure 5: Patient List Page

There is also a button that allows you to view the appointments of a specific patient. When that button is clicked it brings you to another page that allows you to see the doctor/therapist appointments that a patient has booked.

**Specific Patient’s Appointment:** This page contains all appointments for a specific patient. The make payment button allows the receptionist to perform a payment and to update patient’s appointment on behalf of a user.

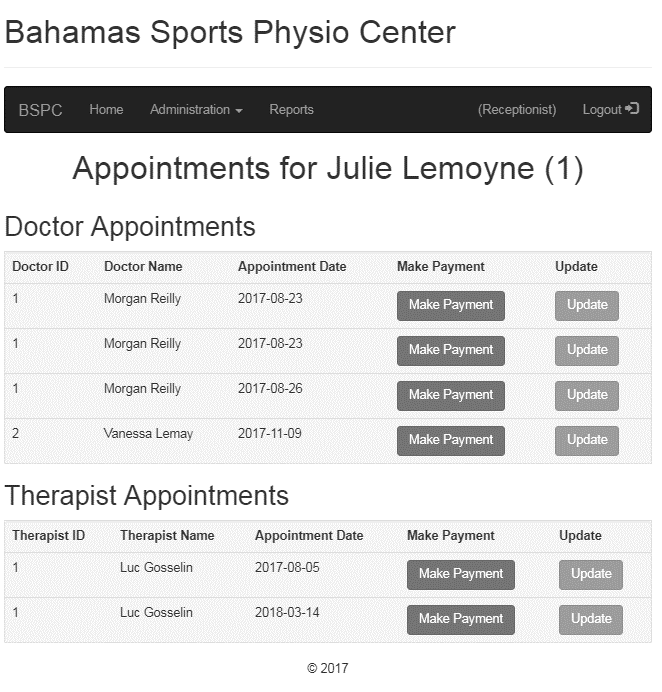


Figure 6: Specific Patient’s Appointment List Page

**Update appointment page:** This is the page that enables a receptionist to reschedule a user’s appointment with a doctor or a therapist.

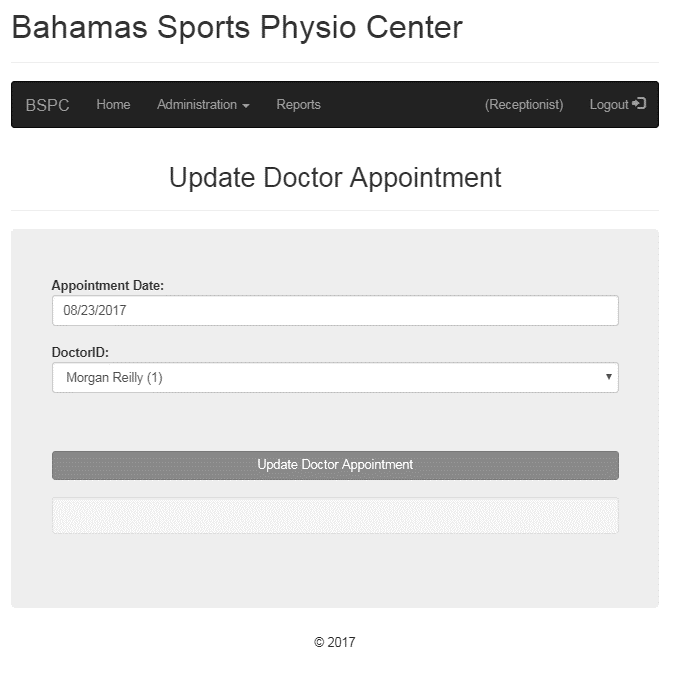


Figure 7: Update Doctor Appointment

**List of Doctor and Therapist**: This page contains the names and IDs of all doctors and therapists.

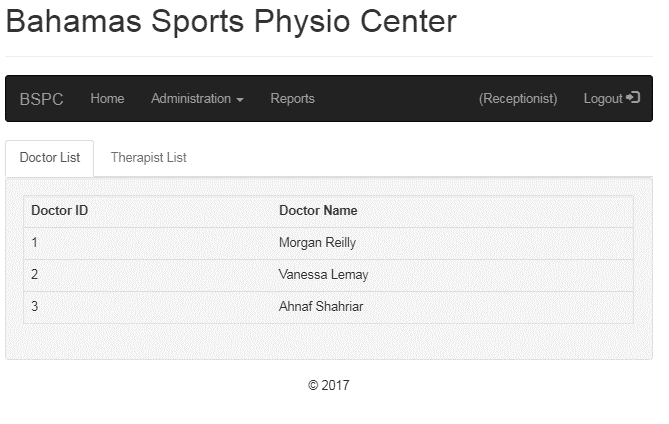


Figure 8: List of Doctor and Therapist

**Reports:** This page displays the number of patients that per therapist between a time lapse. We can also display information about equipment that were never used till today. There is also the possibility to display the list of patients and therapist that have visited the center. There is also the possibility to display the list of therapists that currently work at the center. A receptionist also has the ability to display the reservations made by a patient. The last option that is available on this page is to display doctor and therapist availabilities.

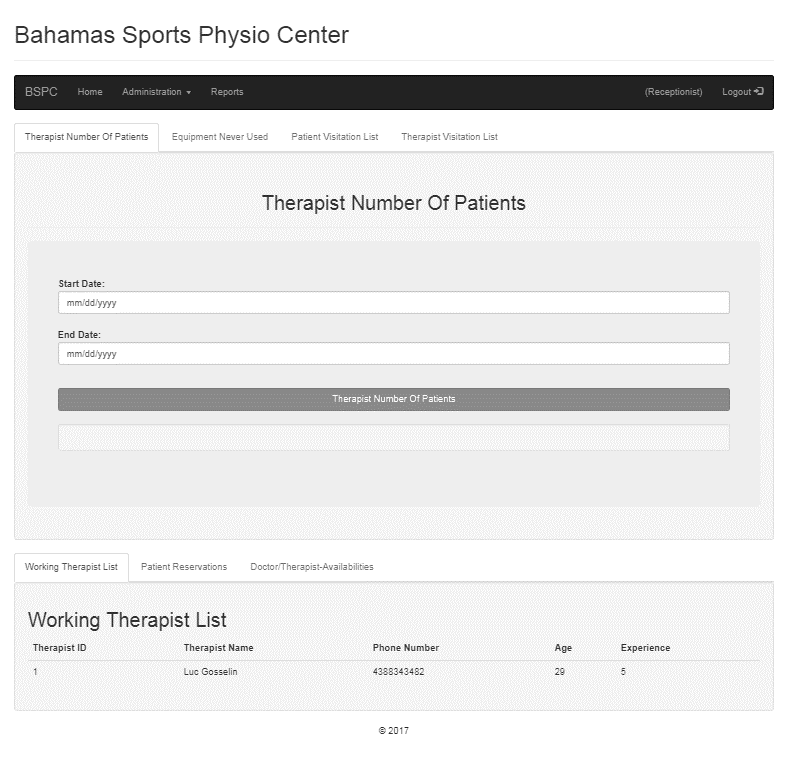


Figure 8: List of Doctor and Therapist

**CONTRIBUTIONS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Jonathan** | **Ahnaf** | **Yeashin** | **Ali** |
| **SQL** | -Created tables  -Queries | -Created tables  -Queries | -Created tables  -Queries | -Created triggers  -Queries |
| **HTML** | All Aspects | All aspects | All aspects | All Aspects |
| **PHP** | -Registration form  -Authentication  -Form generator  -Database connection  -Reviewed  all the code. | -Appointment  -Registration | -Payment  -Header  and footer | -Therapist and doctor appointment page  -Equipment  -Treatment |
| **Documentation** | All Aspects | All Aspects | All Aspects | All Aspects |

**REFERENCES**

[1]"Bootstrap 3 Tutorial", *W3schools.com*, 2017. [Online]. Available: https://www.w3schools.com/bootstrap/. [Accessed: 05- Aug- 2017].

[2]"PHP Connect to MySQL", *W3schools.com*, 2017. [Online]. Available: https://www.w3schools.com/php/php\_mysql\_connect.asp. [Accessed: 05- Aug- 2017].